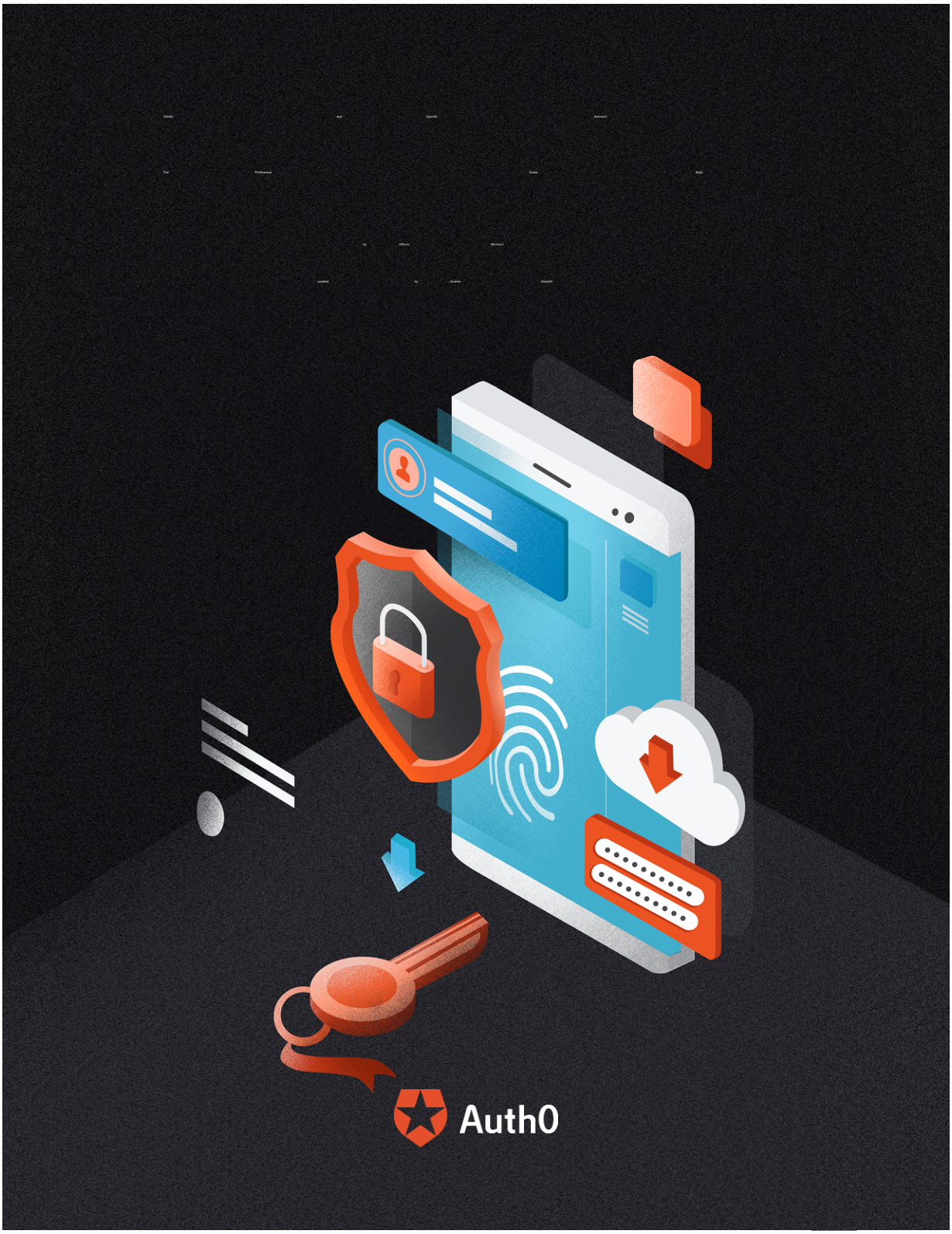
OAuth2 and OpenID Connect: The Professional Guide - *Beta*

by Vittorio Bertocci
curated by Andrea Chiarelli

Auth0


Auth0

OAuth2 and OpenID Connect

The Professional Guide

by William Denniss

created by Andrew Arnott

Introduction 4

Chapter 1 : Introduction to Digital Identity 6

Basic User Passwords in Every App 10

An Overview 10

Cross-Domain SSO 11

The Password Sharing Anti-Pattern 21

Delegated Authentication OAuth2 23

Signing Sign in an app by way of OAuth2 OpenID Connect 27

Auth0 an Identity Signing Complexity at Sig 32

Chapter 2 : OAuth2 and OpenID Connect 38

OAuth2 Basics 41

OAuth2 Scope and Client Basic 48

Chapter 3 : Web Sign In 61

Confidential clients 61

The implicit Grant with front_post 63

A detailed walkthrough 68

Anatomy of an id_token 74

Principles of Token Validation 76

Refreshes and Recovery 78

Chapter 4 : Auding an API from a Web App 82

The Authorization Code Grant 82

Related Dependent Authorization Concepts and Terminology 85

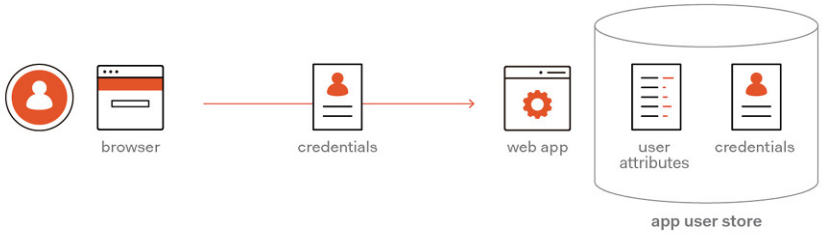
resource

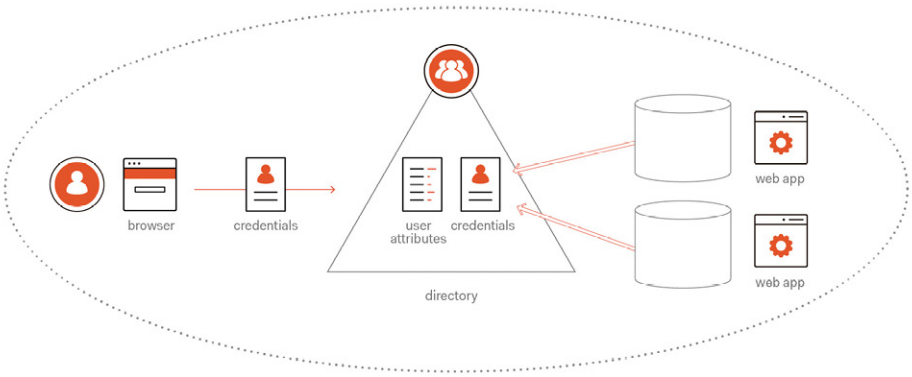
Auth0

- [x]
- [x]
- [x]
- [x]

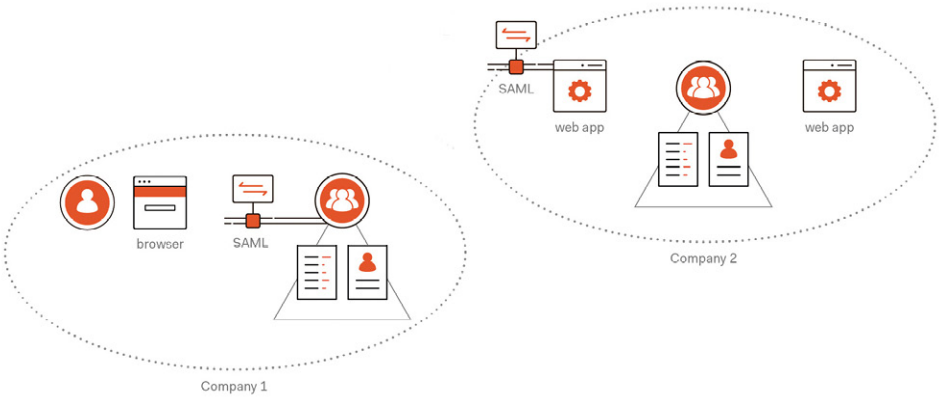
Brian User Passwords in Every App...

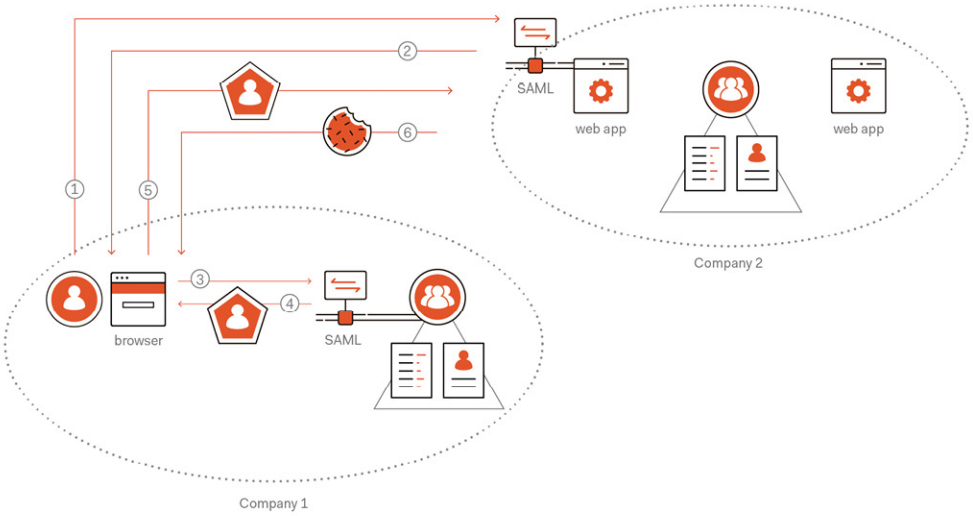
Auth0

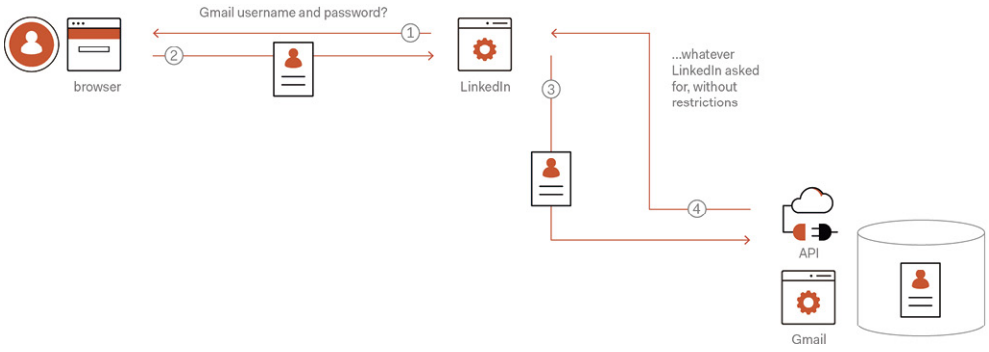

browser
credentials
web app
user attributes
credentials
app user store


browser
credentials
user attributes
credentials
directory
web app
web app

Amos Osman SSO

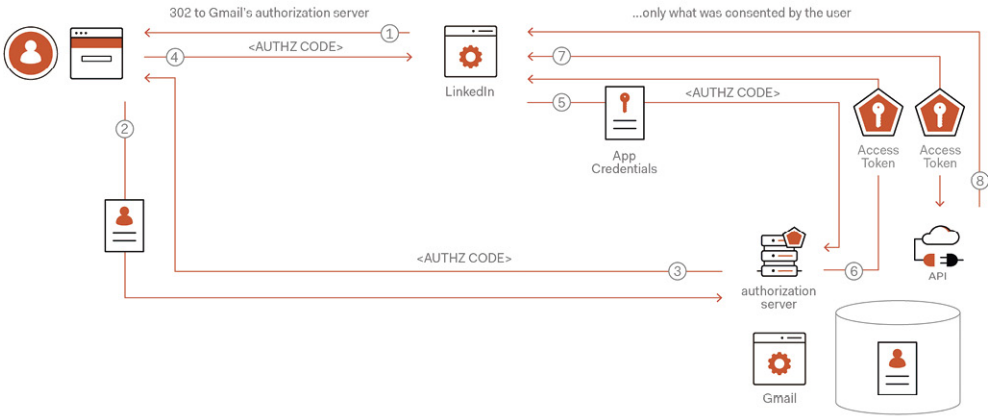

SAML
web app
web app
Company 2
browser
SAML
Company 1


SAML
web app
web app
Company 2
1
2
3
4
5
6
browser
SAML
Company 1

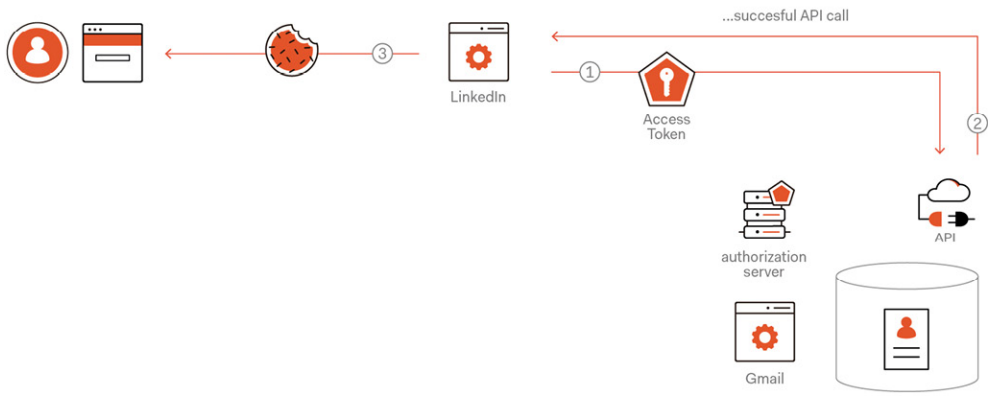

Gmail username and password?
1
2
browser
LinkedIn
3
...whatever
LinkedIn asked
for, without
restrictions
4
API
Gmail

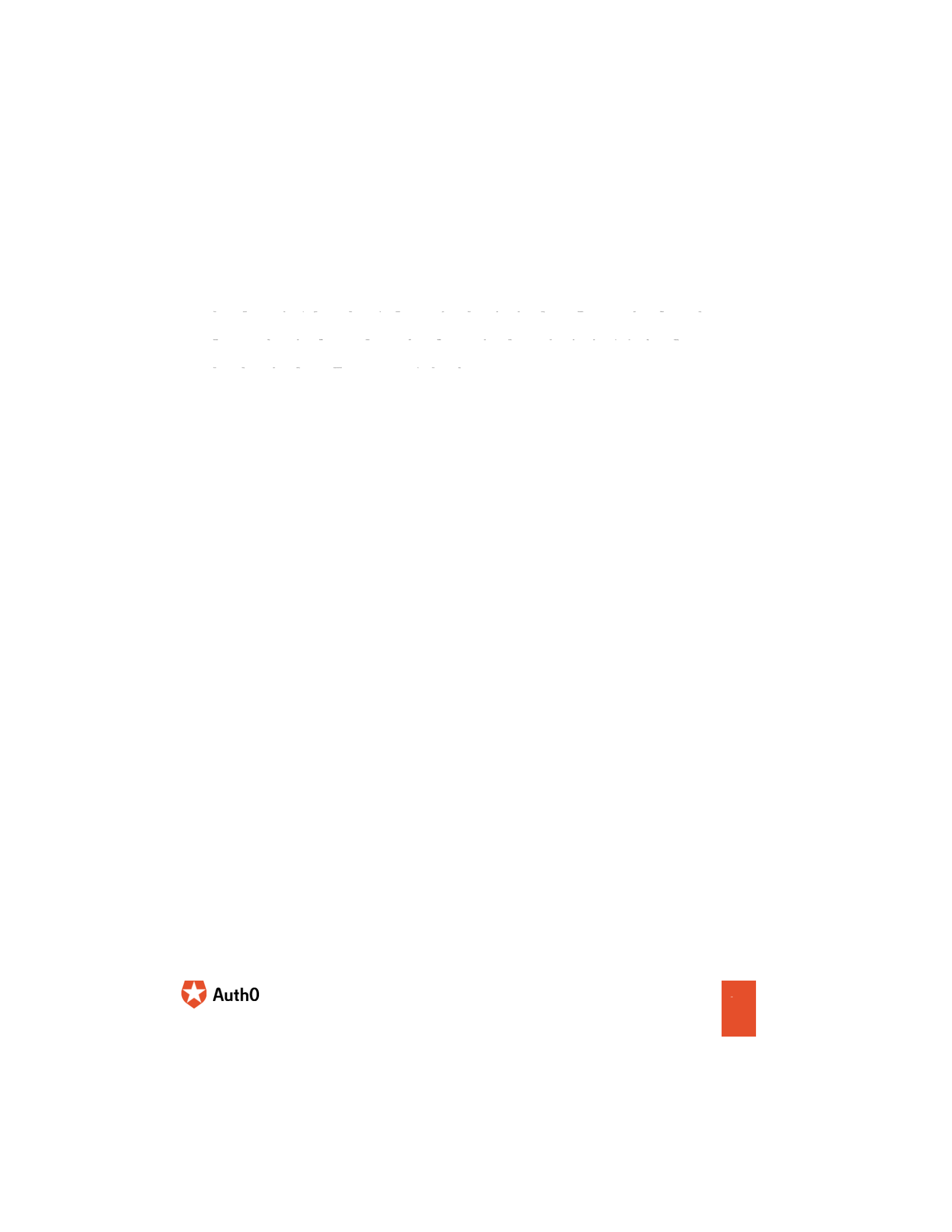
Delegated Authorization OAuth2

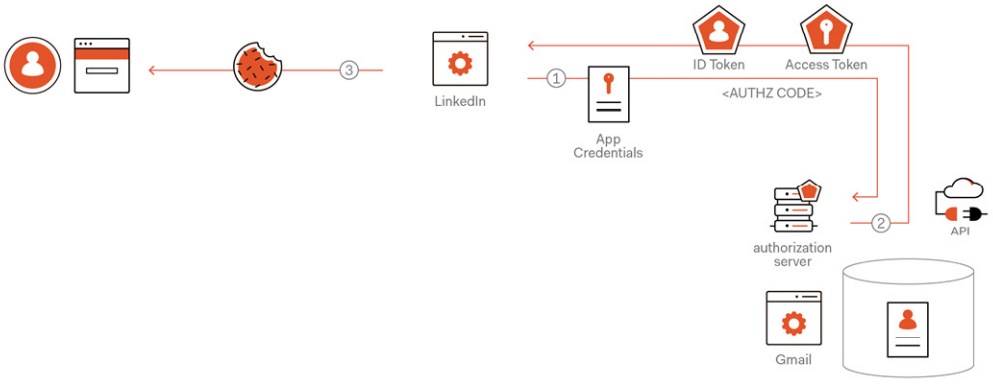
Auth0


302 to Gmail's authorization server
<AUTHZ CODE>
LinkedIn
...only what was consented by the user
<AUTHZ CODE>
App Credentials
Access Token
Access Token
<AUTHZ CODE>
authorization server
API
Gmail

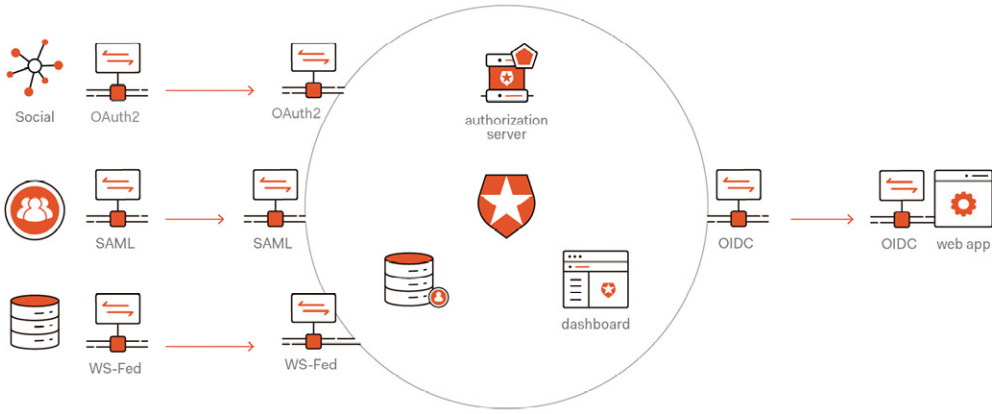
Exploring Sign in on Tap of OAuth2 OpenID Accredi

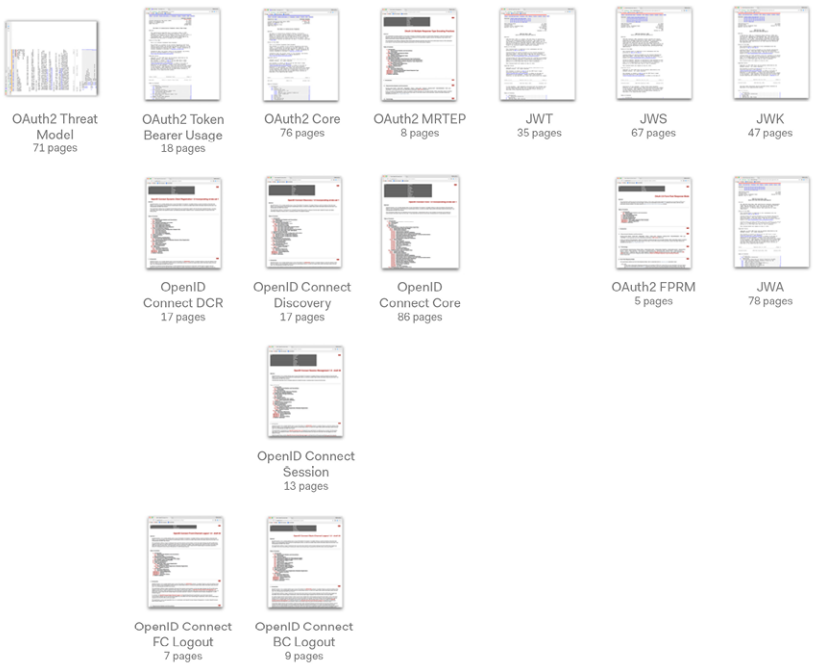

...succesful API call
3
LinkedIn
1
Access
Token
2
API
authorization
server
Gmail




3
LinkedIn
ID Token
Access Token
1
<AUTHZ CODE>
App
Credentials
2
API
authorization
server
Gmail

Auth0


Social
OAuth2
OAuth2
authorization server
SAML
SAML
OIDC
OIDC
web app
dashboard
WS-Fed
WS-Fed

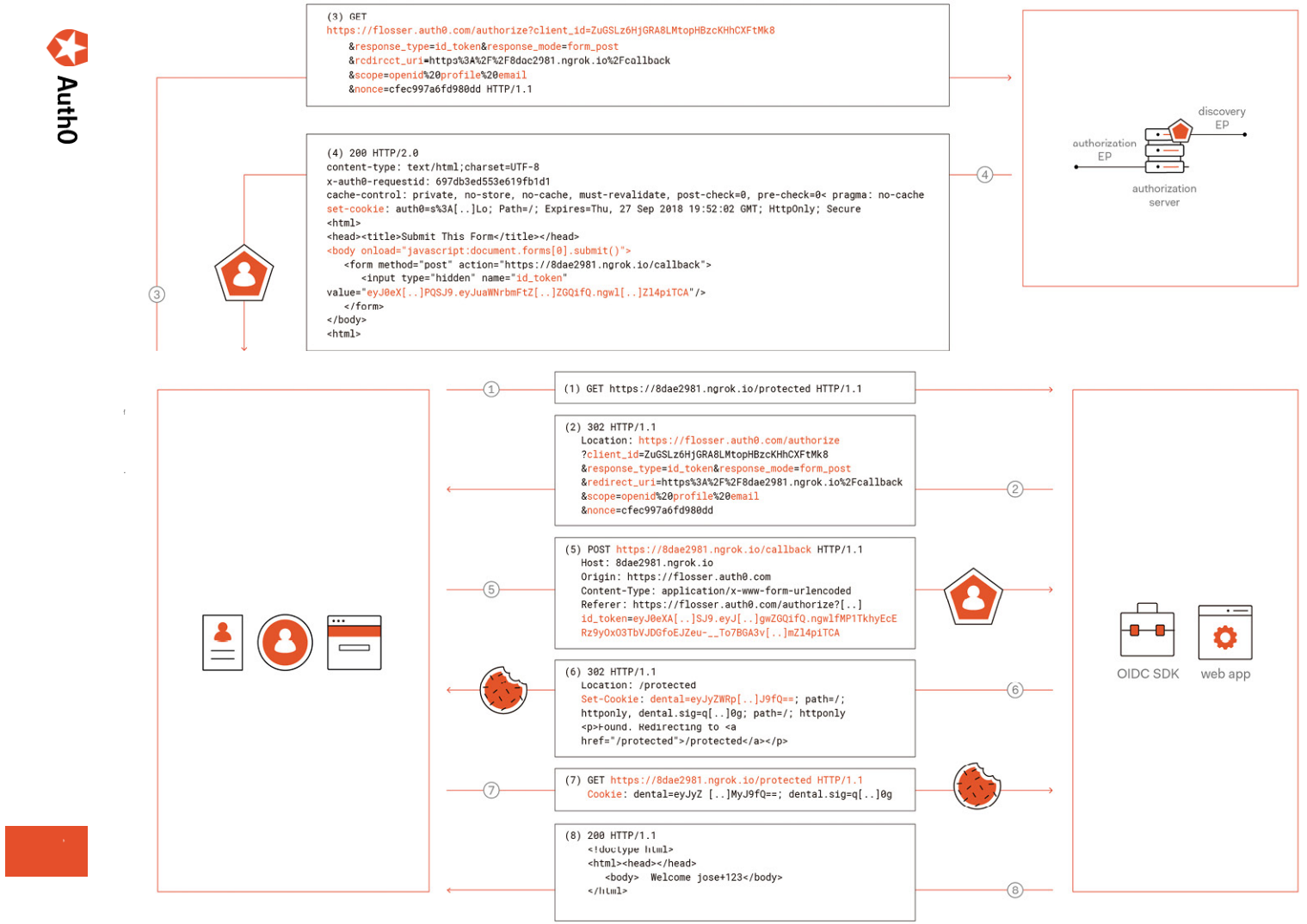

OAuth2 Threat Model
71 pages
OAuth2 Token Bearer Usage
18 pages
OAuth2 Core
76 pages
OAuth2 MRTEP
8 pages
JWT
35 pages
JWS
67 pages
JWK
47 pages
OpenID Connect DCR
17 pages
OpenID Connect Discovery
17 pages
OpenID Connect Core
86 pages
OAuth2 FPRM
5 pages
JWA
78 pages
OpenID Connect Session
13 pages
OpenID Connect FC Logout
7 pages
OpenID Connect BC Logout
9 pages

Auth0

Auth0

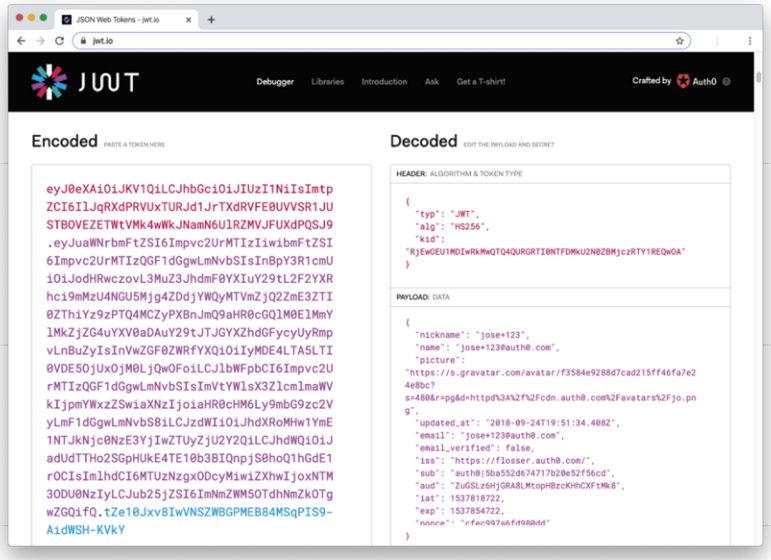
The Implicit Grant with form_post

A detailed walkthrough


Auth0
(3) GET
https://flosser.auth0.com/authorize?client_id=ZuGSLz6HjGRA8LMtopHBzcKHhCXFtMk8
&response_type=id_token&response_mode=form_post
&rdirect_uri=https%3A%2F%2F8dac2981.ngrok.io%2Fcallback
&scope=openid%20profile%20email
&nonce=cfec997a6fd980dd HTTP/1.1
(4) 200 HTTP/2.0
content-type: text/html;charset=UTF-8
x-auth0-requestid: 697db3ed553e619fb1d1
cache-control: private, no-store, no-cache, must-revalidate, post-check=0, pre-check=0< pragma: no-cache
set-cookie: auth0=s%3A[..]Lo; Path=/; Expires=Thu, 27 Sep 2018 19:52:02 GMT; HttpOnly; Secure
<html>
<head><title>Submit This Form</title></head>
<body onload="javascript:document.forms[0].submit()">
<form method="post" action="https://8dae2981.ngrok.io/callback">
<input type="hidden" name="id_token"
value="eyJ0eX[..]PQSJ9.eyJuaWNrbmFtZ[..]ZGQifQ.ngwl[..]Zl4piTCA"/>
</form>
</body>
<html>
discovery EP
authorization EP
authorization server
(1) GET https://8dae2981.ngrok.io/protected HTTP/1.1
(2) 302 HTTP/1.1
Location: https://flosser.auth0.com/authorize
?client_id=ZuGSLz6HjGRA8LMtopHBzcKHhCXFtMk8
&response_type=id_token&response_mode=form_post
&redirect_uri=https%3A%2F%2F8dae2981.ngrok.io%2Fcallback
&scope=openid%20profile%20email
&nonce=cfec997a6fd980dd
(5) POST https://8dae2981.ngrok.io/callback HTTP/1.1
Host: 8dae2981.ngrok.io
Origin: https://flosser.auth0.com
Content-Type: application/x-www-form-urlencoded
Referer: https://flosser.auth0.com/authorize?[..]
id_token=eyJ0eXA[..]SJ9.eyJ[..]gwZGQifQ.ngwlfMP1TkhyEcE
Rz9yOxO3TbVJDGfoEJZeu-__To7BGA3v[..]mZl4piTCA
(6) 302 HTTP/1.1
Location: /protected
Set-Cookie: dental=eyJyZWRp[..]J9fQ==; path=/;
httponly, dental.sig=q[..]0g; path=/; httponly
Found. Redirecting to <a
href="/protected">/protected
(7) GET https://8dae2981.ngrok.io/protected HTTP/1.1
Cookie: dental=eyJyZ [..]MyJ9fQ==; dental.sig=q[..]0g
(8) 200 HTTP/1.1
<!doctype html>
<html><head></head>
<body> Welcome jose+123</body>
</html>
OIDC SDK
web app

the identifier of the app
at the AS
the authorization
endpoint
what artifact(s) I want
how I want the
artifacts returned
(2) 302 HTTP/1.1
Location: https://flosser.auth0.com/authorize
?client_id=ZuGSLz6HjGRA8LMtopHBzcKHhCXFtMk8
&response_type=id_token&response_mode=form_post
&redirect_uri=https%3A%2F%2F8dae2981.ngrok.io%2Fcallback
&scope=openid%20profile%20email
&nonce=cfec997a6fd980dd
where I want to receive
the results back
why I want the artifacts
(what content, capabilities)

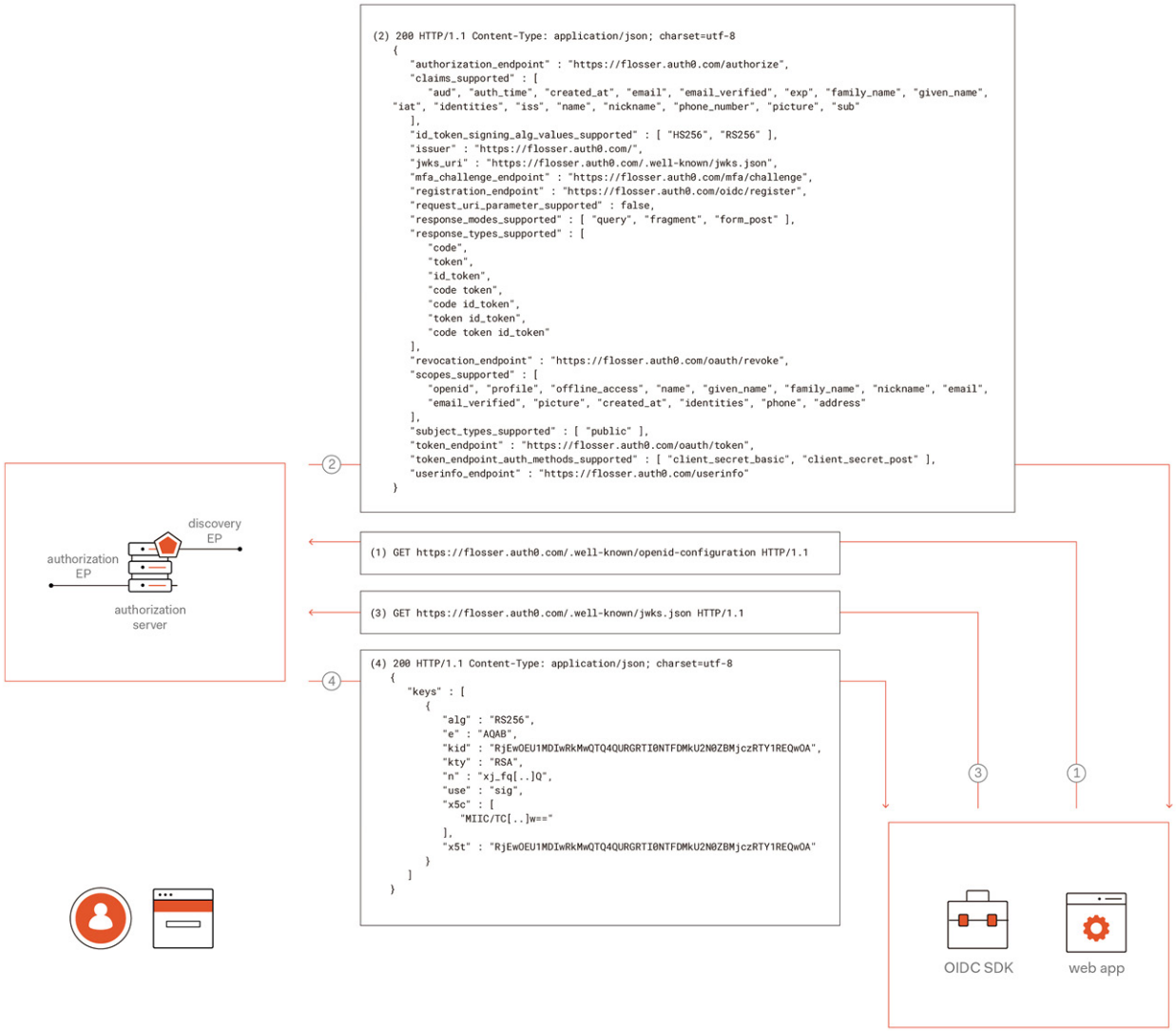

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IlJqRXdPRVUxTURJd1JrTXdRVFE0UVVSR1JUSTBOVEZETWtVMk4wWkJNamN6UlRZMVJFUXdPQSJ9
.eyJuaWNrbmFtZSI6Impvc2UrMTIzIiwibmFtZSI6Impvc2UrMTIzQGF1dGgwLmNvbSIsInBpY3R1cmUiOiJodHRwczovL3MuZ3JhdmF0YXIuY29tL2F2YXR
hci9mMzU4NGU5Mjg4ZDdjYWQyMTVmZjQ2ZmE3ZTI0ZThiYz9zPTQ4MCZyPXBnJmQ9aHR0cHM0ElMkYlMkZjZG4uYXV0aDAuY29tJTJGYXZhdGFycyUyRmp
vLnBuZyIsInVwZGF0ZWRfYXQiOiIyMDE4LTA5LTI0VDE5OjUxOjM0LjQwOFoiLCJlbWFpbCI6Impvc2UrMTIzQGF1dGgwLmNvbSIsImVtYWlsX3Zlcmlma WV
kIjpmYWxzZSwiaXNzIjoiaHR0cHM6Ly9mbG9zc2VyLmF1dGgwLmNvbS8iLCJzdWIiOiJhdXRoMHw1YmE1NTJkNjc0NzE3YjIwZTUyZjU2Y2QiLCJhdWQiOiJ
adUdTTHo2SGpHUKE4TE10b3BIQnpjS0hoQ1hGdE1rOCIsImlhdCI6MTUzNzgxODcyMiwiZXhwIjoxNTM3ODU0NzIyLCJub25jZSI6ImNmZWM5OTdhNmZkOTg
wZGQifQ. ngwlT3mvl3OrvlYYZ02xYhhE-
QGTtf3M_gsJGPtKG5kqhEFWvWHstzWMJE6ZjAapHYitKQg0zDpEbqixaB6PEqz7b09SwLeotZSUBGnfQHU5gC8cpXfMPlTkhyEcERz9yOxO3TbVJDGfoEJZ
eu
__To7BGA3vlvEn7XWHD0Oz45Ht2xtklISfW4vXno_ahZMbLufOJFkpJqtUvHMmd9hVy33uZXp_Z7Vggfk_LDD58XKaJJ8Z9WhPUr1RFll4IPTNEmtmgSEWXz
ds6GYA-Ap5OH2NWIKZe59eDgqi64GPhhjK0u8qSUAue6oQa7M_yw817sJA9yKHdg5mZl4piTCA
```

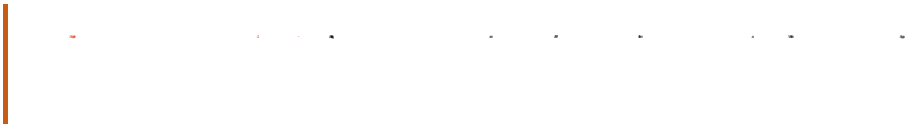

JSON Web Tokens - jwt.io
jwt.io
JWT
Debugger
Libraries
Introduction
Ask
Get a T-shirt!
Crafted by Auth0
Encoded
PASTE A TOKEN HERE
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtp
ZCI6IlJqRXdPRVUxTURJd1JrTXdRVFE0UVVSR1JU
STBOVEZETWtVMk4wWkJNamN6UlRZMVJFUXdPQSJ9
.eyJuaWNrbmFtZSI6Impvc2UrMTIzIiwibmFtZSI
6Impvc2UrMTIzQGF1dGgwLmNvbSIsInBpY3R1cmU
iOiJodHRwczovL3MuZ3JhdmF0YXIuY29tL2F2YXR
hci9mMzU4NGU5Mjg4ZDdjYWQyMTVmZjQ2ZmE3ZTI
0ZThiYz9zPTQ4MCZyPXBnJmQ9aHR0cGQlM0ElMmY
lMkZjZG4uYXV0aDAuY29tJTJGYXZhdGFycyUyRmp
vLnBuZyIsInVwZGF0ZWRfYXQiOiIyMDE4LTA5LTI
0VDE5OjUxOjM0LjQwOFoiLCJlbWFpbCI6Impvc2U
rMTIzQGF1dGgwLmNvbSIsImVtYWlsX3ZlcmlmaWV
kIjpmYWxzZSwiaXNzIjoiaHR0cHM6Ly9mbG9zc2V
yLmF1dGgwLmNvbS8iLCJzdWIiOiJhdXRoMHw1YmE
1NTJkNjc0NzE3YjIwZTUyZjU2Y2QiLCJhdWQiOiJ
adUdTTHo2SGpHUkE4TE10b3BIQnpjS0hoQ1hGdE1
rOCIsImlhdCI6MTUzNzgxODcyMiwiZXhwIjoxNTM
3ODU0NzIyLCJub25jZSI6ImNmZWM5OTdhNmZkOTg
wZGQifQ.tZe10Jxv8IwVNSZWBGPMEB84MSqPIS9-
AidWSH-KVkY
Decoded
EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
{
"typ": "JWT",
"alg": "HS256",
"kid":
"RjEwOEU1MDIwRkMwQTQ4QURGRTI0NTFDMkU2N0ZBMjczRTY1REQwOA"
}
PAYLOAD: DATA
{
"nickname": "jose+123",
"name": "jose+123@auth0.com",
"picture":
"https://s.gravatar.com/avatar/f3584e9288d7cad215ff46fa7e2
4e8bc?
s=480&r=pg&d=https%3A%2f%2Fcdn.auth0.com%2Favatars%2Fjo.pn
g",
"updated_at": "2018-09-24T19:51:34.408Z",
"email": "jose+123@auth0.com",
"email_verified": false,
"iss": "https://flosser.auth0.com/",
"sub": "auth0|5ba552d674717b20e52f56cd",
"aud": "ZuGSLz6HjGRA8LMtopHBzcKHhCXFtMk8",
"iat": 1537818722,
"exp": 1537854722,
"nonce": "cfec997a6fd980dd"
}

Principles of Token Validation

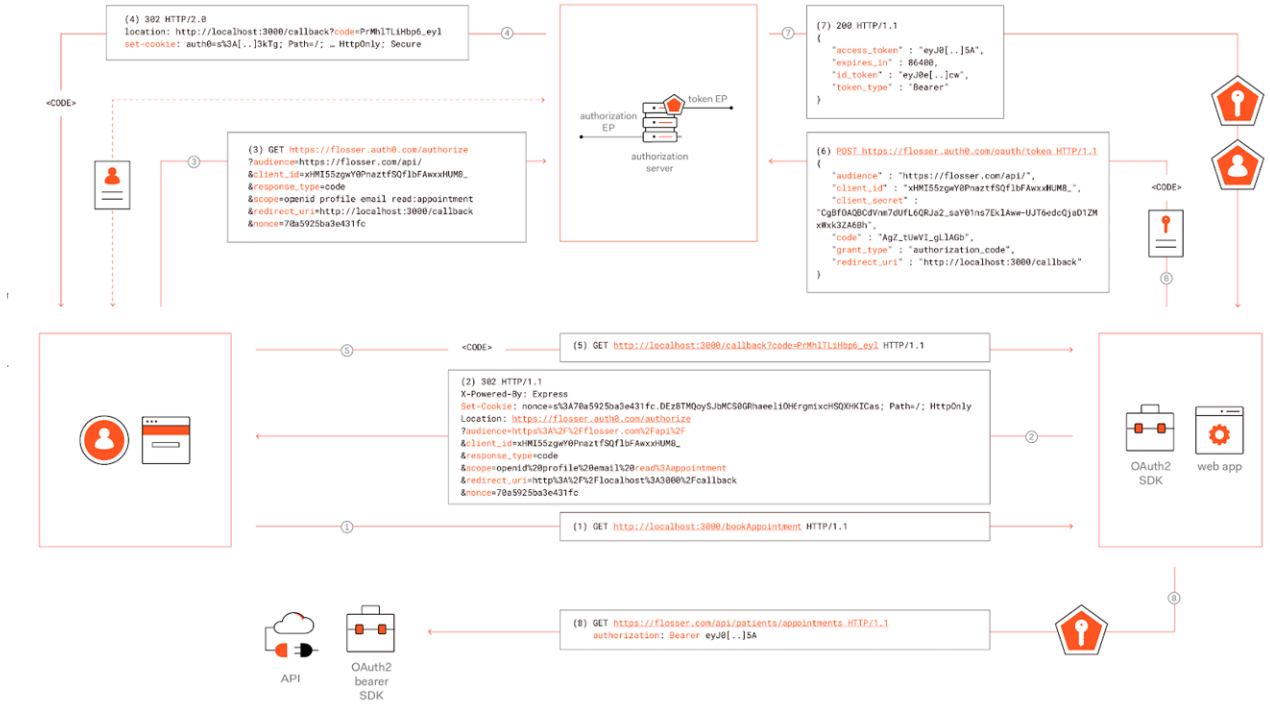
Metadata and Discovery

Auth0

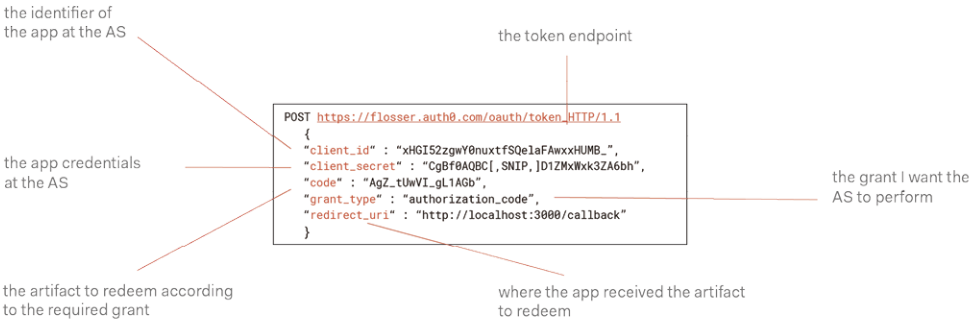

(2) 200 HTTP/1.1 Content-Type: application/json; charset=utf-8
{
"authorization_endpoint" : "https://flosser.auth0.com/authorize",
"claims_supported" : [
"aud", "auth_time", "created_at", "email", "email_verified", "exp", "family_name", "given_name",
"iat", "identities", "iss", "name", "nickname", "phone_number", "picture", "sub"
],
"id_token_signing_alg_values_supported" : ["HS256", "RS256"],
"issuer" : "https://flosser.auth0.com/",
"jwks_uri" : "https://flosser.auth0.com/.well-known/jwks.json",
"mfa_challenge_endpoint" : "https://flosser.auth0.com/mfa/challenge",
"registration_endpoint" : "https://flosser.auth0.com/oidc/register",
"request_uri_parameter_supported" : false,
"response_modes_supported" : ["query", "fragment", "form_post"],
"response_types_supported" : [
"code",
"token",
"id_token",
"code token",
"code id_token",
"token id_token",
"code token id_token"
],
"revocation_endpoint" : "https://flosser.auth0.com/oauth/revoke",
"scopes_supported" : [
"openid", "profile", "offline_access", "name", "given_name", "family_name", "nickname", "email",
"email_verified", "picture", "created_at", "identities", "phone", "address"
],
"subject_types_supported" : ["public"],
"token_endpoint" : "https://flosser.auth0.com/oauth/token",
"token_endpoint_auth_methods_supported" : ["client_secret_basic", "client_secret_post"],
"userinfo_endpoint" : "https://flosser.auth0.com/userinfo"
}
2
discovery EP
authorization EP
authorization server
(1) GET https://flosser.auth0.com/.well-known/openid-configuration HTTP/1.1
(3) GET https://flosser.auth0.com/.well-known/jwks.json HTTP/1.1
(4) 200 HTTP/1.1 Content-Type: application/json; charset=utf-8
{
"keys" : [
{
"alg" : "RS256",
"e" : "AQAB",
"kid" : "RjEwOEU1MDIwRkMwQTQ4QURGRTI0NTFDMkU2N0ZBMjczRTY1REQwOA",
"kty" : "RSA",
"n" : "xj_fq[..]Q",
"use" : "sig",
"x5c" : [
"MIIC/TC[..]w=="
],
"x5t" : "RjEwOEU1MDIwRkMwQTQ4QURGRTI0NTFDMkU2N0ZBMjczRTY1REQwOA"
}
]
}
3
1
OIDC SDK
web app



Auth0


(4) 302 HTTP/2.0
location: http://localhost:3000/callback?code=PrMhlTLiHbp6_eyl
set-cookie: auth0=s%3A[..]3kTg; Path=/; … HttpOnly; Secure
<CODE>
(3) GET https://flosser.auth0.com/authorize
?audience=https://flosser.com/api/
&client_id=xHMI55zgwY0PnaztfSQflbFAwxxHUM8_
&response_type=code
&scope=openid profile email read:appointment
&redirect_uri=http://localhost:3000/callback
&nonce=70a5925ba3e431fc
authorization EP
token EP
authorization server
(7) 200 HTTP/1.1
{
"access_token" : "eyJ0[..]5A",
"expires_in" : 86400,
"id_token" : "eyJ0e[..]cw",
"token_type" : "Bearer"
}
(6) POST https://flosser.auth0.com/oauth/token HTTP/1.1
{
"audience" : "https://flosser.com/api/",
"client_id" : "xHMI55zgwY0PnaztfSQflbFAwxxHUM8_",
"client_secret" :
"CgBfOAQBCdVnm7dUfL6QRJa2_saY01ns7EklAww-UJT6edcQjaD1ZMxMxk3ZA6Bh",
"code" : "AgZ_tUwYI_gL1AGb",
"grant_type" : "authorization_code",
"redirect_uri" : "http://localhost:3000/callback"
}
<CODE>
<CODE>
(5) GET http://localhost:3000/callback?code=PrMhlTLiHbp6_eyl HTTP/1.1
(2) 302 HTTP/1.1
X-Powered-By: Express
Set-Cookie: nonce=s%3A70a5925ba3e431fc.DEz8TMQoySJbMCS0GRhaeeliOHfrgmixcHSQXHKICas; Path=/; HttpOnly
Location: https://flosser.auth0.com/authorize
?audience=https%3A%2F%2Fflosser.com%2Fapi%2F
&client_id=xHMI55zgwY0PnaztfSQflbFAwxxHUM8_
&response_type=code
&scope=openid%20profile%20email%20read%3Aappointment
&redirect_uri=http%3A%2F%2Flocalhost%3A3000%2Fcallback
&nonce=70a5925ba3e431fc
(1) GET http://localhost:3000/bookAppointment HTTP/1.1
OAuth2 SDK
web app
(8) GET https://flosser.com/api/patients/appointments HTTP/1.1
authorization: Bearer eyJ0[..]SA
API
OAuth2 bearer SDK

- ✓
- ✓
- ✓
- ✓


the identifier of
the app at the AS
the token endpoint
the app credentials
at the AS
POST https://flosser.auth0.com/oauth/token HTTP/1.1
{
"client_id" : "xHGI52zgwY0nuxtfSQelaFAwxxHUMB_",
"client_secret" : "CgBf0AQBC[,SNIP,]D1ZMxWxk3ZA6bh",
"code" : "AgZ_tUwVI_gL1AGb",
"grant_type" : "authorization_code",
"redirect_uri" : "http://localhost:3000/callback"
}
the grant I want the
AS to perform
the artifact to redeem according
to the required grant
where the app received the artifact
to redeem

Auth0

Auth0

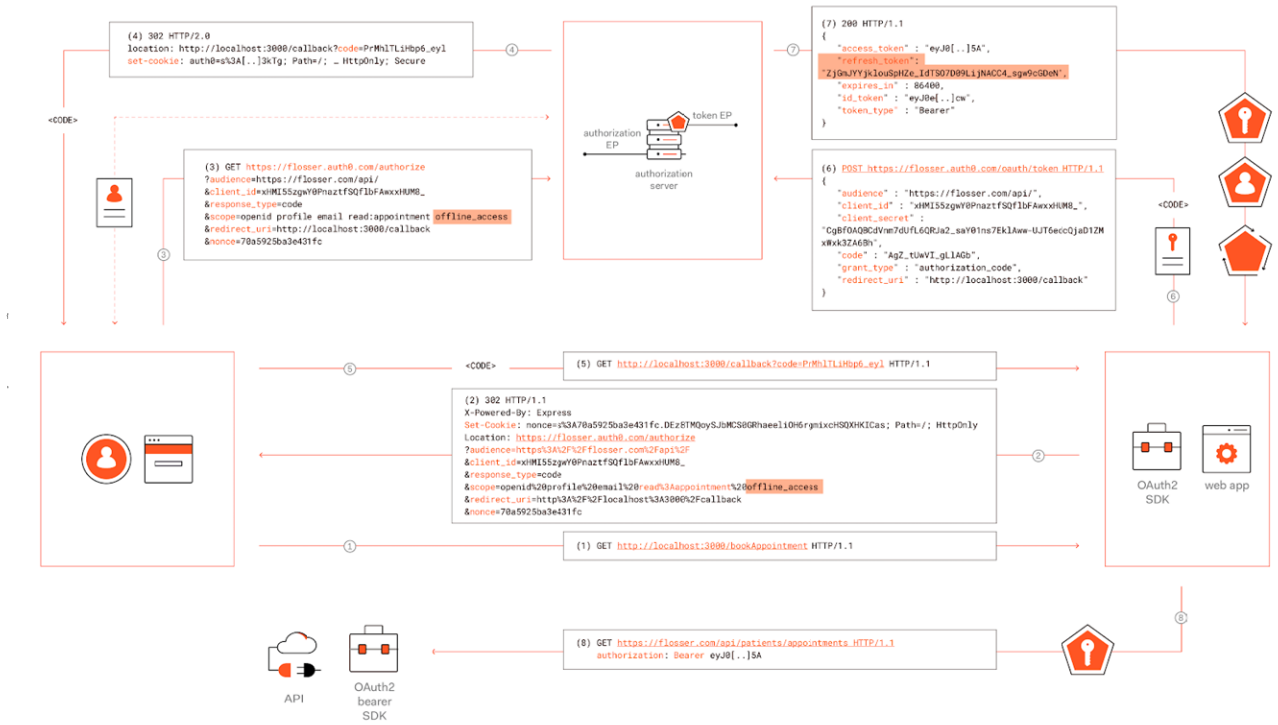
Authorization Code Grant and PKCE

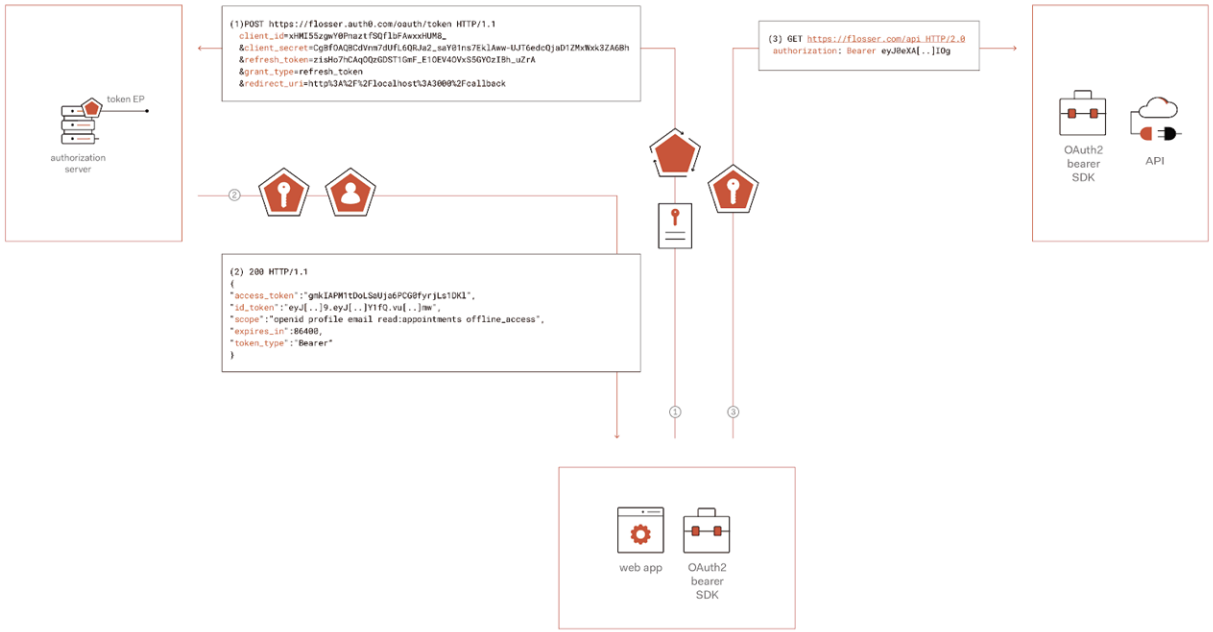
Status Overlaid Authorization Accepts and Tendingy

Permissions

Privileges

Auth0


(4) 302 HTTP/2.0
location: http://localhost:3000/callback?code=PrMhlTLiHbp6_eyl
set-cookie: auth0=s%3A[..]3kTg; Path=/; … HttpOnly; Secure
<CODE>
(3) GET https://flosser.auth0.com/authorize
?audience=https://flosser.com/api/
&client_id=xHMI55zgwY0PnaztfSQflbFAwxxHUM8_
&response_type=code
&scope=openid profile email read:appointment offline_access
&redirect_uri=http://localhost:3000/callback
&nonce=70a5925ba3e431fc
authorization EP
token EP
authorization server
(7) 200 HTTP/1.1
{
"access_token" : "eyJ0[..]5A",
"refresh_token":
"ZjGmJYYjklouSpHZe_IdTSO7D09LijNACC4_sgw9cGDeN",
"expires_in" : 86400,
"id_token" : "eyJ0e[..]cw",
"token_type" : "Bearer"
}
(6) POST https://flosser.auth0.com/oauth/token HTTP/1.1
{
"audience" : "https://flosser.com/api/",
"client_id" : "xHMI55zgwY0PnaztfSQflbFAwxxHUM8_",
"client_secret" :
"CgBfOAQ0CdVnm7dUfL6QRJa2_saY01ns7Ek1Aww-UJT6eccQjaD1ZM
x0xk3ZA6Bh",
"code" : "AgZ_tUwVI_gL1AGb",
"grant_type" : "authorization_code",
"redirect_uri" : "http://localhost:3000/callback"
}
<CODE>
<CODE>
(5) GET http://localhost:3000/callback?code=PrMhlTLiHbp6_eyl HTTP/1.1
(2) 302 HTTP/1.1
X-Powered-By: Express
Set-Cookie: nonce=s%3A70a5925ba3e431fc.DEz8TMQoySJbMCS8GRhaeeliOH6rgmixcHSQXHKICas; Path=/; HttpOnly
Location: https://flosser.auth0.com/authorize
?audience=https%3A%2F%2Fflosser.com%2Fapi%2F
&client_id=xHMI55zgwY0PnaztfSQflbFAwxxHUM8_
&response_type=code
&scope=openid%20profile%20email%20read%3Aappointment%20offline_access
&redirect_uri=http%3A%2F%2Flocalhost%3A3000%2Fcallback
&nonce=70a5925ba3e431fc
(1) GET http://localhost:3000/bookAppointment HTTP/1.1
OAuth2 SDK
web app
API
OAuth2 bearer SDK
(8) GET https://flosser.com/api/patients/appointments HTTP/1.1
authorization: Bearer eyJ0[..]5A


token EP
authorization server
(1)POST https://flosser.auth0.com/oauth/token HTTP/1.1
client_id=xHMI55zgwY0PnaztfSQflbFAwxxHUM8_
&client_secret=CgBfOAQBCdVnm7dUfL6QRJa2_saY01ns7EklAww-UJT6edcOjaD1ZMxNxk3ZA6Bh
&refresh_token=zisHo7hCAqOQzGDST1GmF_E1OEV4OVx55GYOzIBh_uZrA
&grant_type=refresh_token
&redirect_uri=http%3A%2F%2Flocalhost%3A3000%2Fcallback
(3) GET https://flosser.com/api HTTP/2.0
authorization: Bearer eyJ0eXA[..]IOg
OAuth2 bearer SDK
API
(2) 200 HTTP/1.1
{
"access_token":"gmkIAPM1tDoLSaUja6PCG0fyrjLs1DK1",
"id_token":"eyJ[..]9.eyJ[..]Y1fQ.vu[..]mw",
"scope":"openid profile email read:appointments offline_access",
"expires_in":86400,
"token_type":"Bearer"
}
web app
OAuth2 bearer SDK

Sidebar: Access Tokens vs. ID Tokens

Access Tokens Recap

Auth0

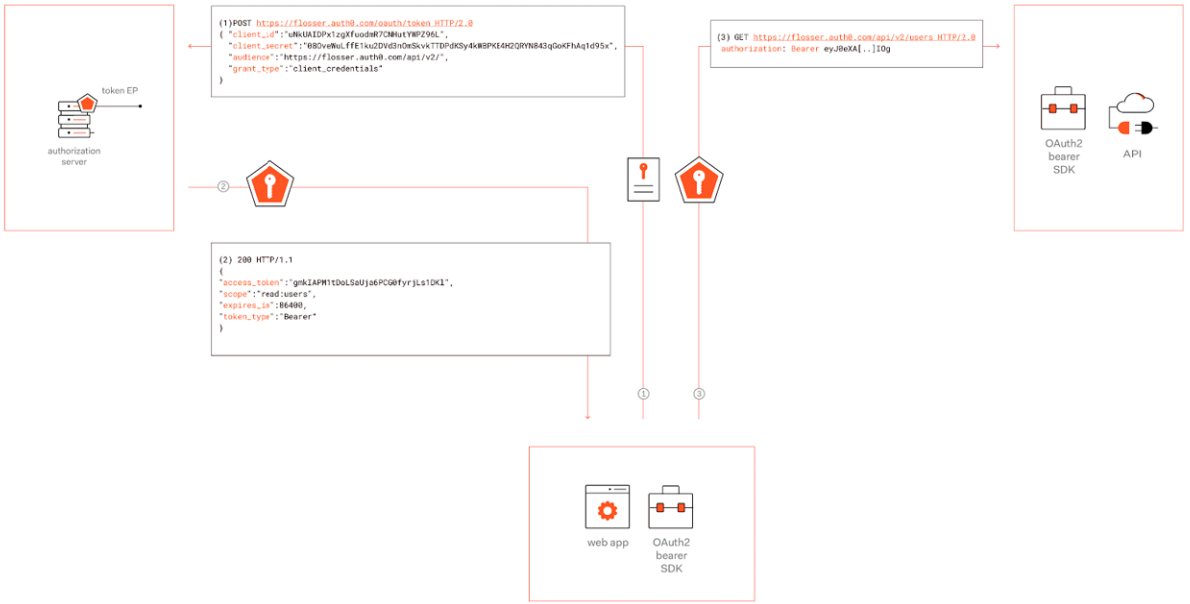
ID Tokens and the Back Channel

Auth0

userinfo EP
authorization server
(2) 200 HTTP/2.0
{ "sub": "auth0|5ba552d674717b20e52f56cd",
"nickname": "jose+123",
"name": "jose+123@auth0.com",
"picture": "https://s.grav[..]%2Fjo.png",
"updated_at": "2018-09-24T19:51:34.408Z",
"email": "jose+123@auth0.com",
"email_verified": false
}
(1) GET https://flosser.auth0.com/userinfo HTTP/2.0
authorization: Bearer dPvEBYtacnAlDeR4Kn0NIpwQ70i81bJh
web app

Auth0

Auth0


token EP
authorization server
(1)POST https://flosser.auth0.com/oauth/token HTTP/2.0
{ "client_id":"sNkUAIDPv1zgXfuodsH7CNHutYWPZ9AL",
"client_secret":"B8OveWuLffE1ku2DVd3nOmSkvkTTDPdKSy4kWBPKE4H2QRYN843qGoKFhAq1d95x",
"audience":"https://flosser.auth0.com/api/v2/",
"grant_type":"client_credentials"
}
(2) 200 HTTP/1.1
{
"access_token":"gmkIAPM1tDoLSbUjo6PCG0fyrjLs1DK1",
"scope":"read:users",
"expires_in":86400,
"token_type":"Bearer"
}
(3) GET https://flosser.auth0.com/api/v2/users HTTP/2.0
authorization: Bearer eyJ0eXA[..]IOg
OAuth2 bearer SDK
API
web app
OAuth2 bearer SDK

Auth0

Auth0

Auth0

Auth0